



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/822,431	04/12/2004	Guillermo Roa	112056-0164	2566
24267 7590 07/07/2009 CESARI AND MCKENNA, LLP 88 BLACK FALCON AVENUE BOSTON, MA 02210				
EXAMINER DUONG, THOMAS				
ART UNIT PAPER NUMBER 2445				
MAIL DATE DELIVERY MODE 07/07/2009 PAPER				

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/822,431

Applicant(s)

ROA, GUILLERMO

Examiner

Thomas Duong

Art Unit

2445

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 16 March 2009.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-13, 15-20 and 24-26 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-13, 15-20 and 24-26 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 12 April 2004 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB08)
Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

DETAILED ACTION

Response to Amendment

1. This office action is in response to the Applicant's After Non-Final Amendment filed on March 16, 2009. Applicant amended *claims 1, 16, 20 and 24-26*. *Claims 1-13, 15-20 and 24-26* are presented for further consideration and examination.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.
3. *Claims 1-3, 6, 12-13, 15-17, 20, and 24-26* are rejected under 35 U.S.C. 103(a) as being unpatentable over Becker-Szendy et al. (US007243089B2) and in view of Kazar et al. (US006868417B2).
4. With regard to *claims 1, 16, 20, and 24-26*, Becker-Szendy discloses,
 - *a plurality of network resources adapted to process received block-based protocol data access requests; and* (Becker-Szendy, col.1, line 10 – col.20, line 48)Becker-Szendy discloses, *"The present invention satisfies this need, and presents a system, a computer program product, and an associated method*

(collectively referred to herein as "the system" or "the present system") and a service for federating a local file system into a distributed file system (FS), while preserving local access to the existing data in the local file system. The present system may provide indirect access to local file systems using protocols such as, for example, storage tank protocols, object-based storage protocols, block-based protocols, etc. The server-based design of the present system allows systems to migrate their data and share the management of data. The data is federated, or made available to various clients by making it on-line to each client. The present system may be used with any file system protocol that supports migration, consistency and multi-host federation" (Becker-Szendy, col.2, lines 49-64).

Becker-Szendy discloses, "For purposes of illustration, the use of the present system is described in terms of a storage tank system. Storage tank is a distributed file system built on a storage area network. Data may be stored either in block storage devices or object storage servers. Unlike most file systems, meta-data and data are stored separately in the storage tank system. The server manages meta-data comprising the location of the blocks of each file/object on shared storage. Object storage servers enable the creation of self-managed, heterogeneous, shared storage by offering a higher-level storage abstraction in the form of objects" (Becker-Szendy, col.2, line 65 – col.3, line 8). Becker-Szendy discloses, "The virtual object storage server 250 and the virtual metadata server 245 run against the local file system 210. The storage tank client 235 sends metadata requests to the virtual metadata server 245" (Becker-Szendy, col.10, lines 38-41). Hence, Becker-Szendy teaches of the system (e.g., storage tank system, distributed file system built on a storage area network) providing

indirect access (i.e., Applicant's adapted to process) to local file systems (i.e., Applicant's plurality of network resources) using protocols such as storage tank protocols, object-based storage protocols, block-based protocols, etc. (i.e., Applicant's one or more block-based protocols) by having the storage tank client sending metadata requests (i.e., Applicant's data access request) and the virtual metadata server responding to the requests appropriately.

- a plurality of virtual servers each comprising a logical partitioning of the network resources to establish an instance of a multi-protocol server configured to service the block-based data access requests by converting the blocked-based protocol requests to appropriate file system data requests, each virtual server allowed shared access to resources of the file system; and (Becker-Szendy, col.1, line 10 – col.20, line 48)

Becker-Szendy discloses, *"An embodiment of the federation design of the present system relies on object storage servers. The present system creates a virtual storage tank server and a virtual object storage server on top of the local file system to make the local file system appear as both a storage tank node and an object based storage server to a storage tank system. Data accesses go through the virtual object storage server, and not through the virtual storage tank server. It should be clear that the storage tank server provides metadata information to the client, who then accesses the data directly from the object based storage server. Storage tank uses the object storage server interface to access the local file system data. After the local file system is exposed through the storage tank file system, data may be left on-line and stored in the local file system or migrated to a new storage device through storage tank tools. While the*

present system is described in terms of storage tank, it is not limited to storage tank and may be expanded to other file systems" (Becker-Szendy, col.3, lines 50-67). Hence, Becker-Szendy teaches of the virtual storage tank server and the virtual object storage server (i.e., Applicant's one or more vfilers) on top of (i.e., Applicant's comprising a logical partitioning) the local file systems (i.e., Applicant's network resources). Becker-Szendy teaches of the virtual object storage server (i.e., Applicant's multi-protocol server) allowing (i.e., Applicant's configured to service) data accesses (i.e., Applicant's data access requests) indirectly to the local file systems (i.e., Applicant's network resources) using (in response to) protocols such as block-based protocol. Becker-Szendy discloses, *"In the local file system 210, objects are accessed not by their object ID number but by their file names. System 205 provides a mapping from file name to object ID number, and a reverse mapping from object ID number back to the file name. This mapping is unambiguous and stable. System 205 implements the mapping by having the virtual metadata server 245 and the virtual object storage server 250 use the object ID database 255 that is shared between them. The object ID database 255 maps the file name to object ID number, and reverse maps the object ID number to a file name"* (Becker-Szendy, col.11, lines 1-11). Hence, Becker-Szendy teaches of the system 205, which includes the virtual metadata server 245 and the virtual object storage server 250 (i.e., Applicant's virtual servers), implementing the mapping (i.e., Applicant's converting) of the client's data access requests so that they can be understood by local file system 210 (i.e., Applicant's to appropriate file system data requests).

- *a context data structure provided to each virtual server, the context data structure including information pertaining to a security domain of the virtual server for each supported access protocol, to enable controlled access to the shared resources of the file system.* (Becker-Szendy, col.1, line 10 – col.20, line 48; Kazar, col.1, line 7 – col.11, line 19)

Becker-Szendy discloses, “File data is the bytes that are actually stored in a file. Metadata is all the rest of the information stored in a file system. Metadata comprises the directory tree and the attributes of objects such as files and directories. The directory tree is a set of names that are arranged in directories, forming a tree structure. Typical attributes comprise time stamps (i.e., time created, time last modified, time last read) and security related attributes (i.e., the identity of the owner of the object and a description of what the owner or other parties may do to the object). In addition, with object based storage, it is possible to have some of the metadata stored on the object based storage. For example, object based storage manages the block mapping internally, off-loading this role from the file system” (Becker-Szendy, col.7, lines 41-54).

However, Becker-Szendy does not explicitly disclose,

- *a plurality of network resources adapted to process received block-based protocol data access requests; and*
- *a plurality of virtual servers each comprising a logical partitioning of the network resources to establish an instance of a multi-protocol server configured to service the block-based data access requests by converting the blocked-based protocol requests to appropriate file system data requests, each virtual server allowed shared access to resources of the file system; and*

Kazar teaches,

- *a plurality of network resources adapted to process received block-based protocol data access requests; and* (Kazar, col.1, line 7 – col.11, line 19)

Kazar discloses, “The block_login operation passes in a user ID and a password, and authenticates the user for the service. Based upon the user, the server chooses a particular file system to which the user’s block read and write operations will be applied. This file system must be a SAN file system, which means that it contains one file, whose blocks corresponding directly to the block addresses read or written by the block service client” (Kazar, col.9, line 64 – col.10, line 4). Hence, Kazar teaches of the server (i.e., Applicant’s network resource) choosing (i.e., Applicant’s processing) the user’s block read operation (i.e., Applicant’s received block-based protocol data access request).

- *a plurality of virtual servers each comprising a logical partitioning of the network resources to establish an instance of a multi-protocol server configured to service the block-based data access requests by converting the blocked-based protocol requests to appropriate file system data requests, each virtual server allowed shared access to resources of the file system; and* (Kazar, col.1, line 7 – col.11, line 19)

Kazar discloses, “The block_login operation passes in a user ID and a password, and authenticates the user for the service. Based upon the user, the server chooses a particular file system to which the user’s block read and write operations will be applied. This file system must be a SAN file system, which means that it contains one file, whose blocks corresponding directly to the block

addresses read or written by the block service client" (Kazar, col.9, line 64 – col.10, line 4). Hence, Kazar teaches of the server (i.e., Applicant's network resource) choosing (i.e., Applicant's processing) the user's block read operation (i.e., Applicant's received block-based protocol data access request) by converting (implied) the requests to the particular file system (i.e., Applicant's appropriate file system) based upon the user.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of Kazar with the teachings of Becker-Szendy to provide *"a system, a computer program product, and an associated method (collectively referred to herein as "the system" or "the present system") and a service for federating a local file system into a distributed file system (FS), while preserving local access to the existing data in the local file system. The present system may provide indirect access to local file systems using protocols such as, for example, storage tank protocols, object-based storage protocols, block-based protocols, etc. The server-based design of the present system allows systems to migrate their data and share the management of data. The data is federated, or made available to various clients by making it on-line to each client. The present system may be used with any file system protocol that supports migration, consistency and multi-host federation"* (Becker-Szendy, col.2, lines 50-64). Kazar discloses, *"The present invention pertains to an apparatus for handling file level and block level remote file accesses. The apparatus comprises a block level server. The apparatus comprises a file level server. The apparatus comprises a storage layer implementing an inode layer performing inode operations, and storing data accessed by the file level and block level servers. The apparatus comprises a management*

layer connected to the storage layer underlying the block and file level servers, which performs data management operations upon the underlying data” (Kazar, col.1, lines 29-38).

5. With regard to claim 2, Becker-Szendy and Kazar disclose,

- *wherein the network resources comprise network interfaces assigned to one or more network address resources. (Becker-Szendy, col.1, line 10 – col.20, line 48; Kazar, col.1, line 7 – col.11, line 19)*

Becker-Szendy discloses, “Storage tank uses the object storage server interface to access the local file system data. After the local file system is exposed through the storage tank file system, data may be left on-line and stored in the local file system or migrated to a new storage device through storage tank tools. While the present system is described in terms of storage tank, it is not limited to storage tank and may be expanded to other file systems” (Becker-Szendy, col.3, lines 50-67).

6. With regard to claims 3 and 17 Becker-Szendy and Kazar disclose,

- *further comprising storage media configured to store information as units of storage resources, the units of storage resources allocated among each of the vfilers. (Becker-Szendy, col.1, line 10 – col.20, line 48; Kazar, col.1, line 7 – col.11, line 19)*

Becker-Szendy discloses, “The term “disk” references the actual storage device, whether it is an individual disk (attached via DAS or SAN), or a logical unit on a disk array. Disks that are used by the storage tank system 300 are attached via a

SAN. The SAN may be comprised of, for example, fiber channel, iSCSI, etc. A disk that partitions its storage space into many objects is an object store device, also referenced as an object based store. The traditional disk is called a block disk. The storage tank system 300 can use both traditional block disks, as well as object store devices to store the contents of files" (Becker-Szendy, col.3, lines 50-67). Hence, Beck-Szendy teaches of the actual storage device (i.e., Applicant's storage media) partitioned (i.e., Applicant's configured to store information) its storage space into many object store devices (i.e., Applicant's units of storage resources). Becker-Szendy teaches the storage tank system, which includes the virtual storage tank server and the virtual object storage server (i.e., Applicant's vfilers), using (i.e., Applicant's allocated) both traditional block disks and object store devices (i.e., Applicant's units of storage resources).

7. With regard to claim 6, Becker-Szendy and Kazar disclose,

- *further comprising an operating system having a file system resource adapted to perform a boundary check to verify that a request is allowed to access certain units of the storage resources on the storage media, each vfiler allowed shared access to the file system and further adapted to create virtual disks within the units of storage resources and wherein each of the virtual disks associated with one or more of the vfilers.* (Becker-Szendy, col.1, line 10 – col.20, line 48; Kazar, col.1, line 7 – col.11, line 19)

Becker-Szendy discloses, *"Data consistency is maintained in that existing applications may modify data in the file system during migration or federation. During federation, other computer systems (or hosts) may modify the data in the*

file system if access control information allows them to do so. All changes in the file system are seen consistently on all hosts. Minimal downtime is required to install the present system and reconfigure the existing applications to communicate with the present system" (Becker-Szendy, col.3, lines 20-28).

8. With regard to claims 12-13, Becker-Szendy and Kazar disclose,

- *wherein the block-based protocol comprises iSCSI.* (Becker-Szendy, col.1, line 10 – col.20, line 48; Kazar, col.1, line 7 – col.11, line 19)

Becker-Szendy discloses, *"The term "disk" references the actual storage device, whether it is an individual disk (attached via DAS or SAN), or a logical unit on a disk array. Disks that are used by the storage tank system 300 are attached via a SAN. The SAN may be comprised of, for example, fiber channel, iSCSI, etc. A disk that partitions its storage space into many objects is an object store device, also referenced as an object based store. The traditional disk is called a block disk. The storage tank system 300 can use both traditional block disks, as well as object store devices to store the contents of files"* (Becker-Szendy, col.3, lines 50-67).

- *wherein the block-based protocol comprises FCP.* (Becker-Szendy, col.1, line 10 – col.20, line 48; Kazar, col.1, line 7 – col.11, line 19)

Becker-Szendy discloses, *"The term "disk" references the actual storage device, whether it is an individual disk (attached via DAS or SAN), or a logical unit on a disk array. Disks that are used by the storage tank system 300 are attached via a SAN. The SAN may be comprised of, for example, fiber channel, iSCSI, etc. A disk that partitions its storage space into many objects is an object store device,*

also referenced as an object based store. The traditional disk is called a block disk. The storage tank system 300 can use both traditional block disks, as well as object store devices to store the contents of files" (Becker-Szendy, col.3, lines 50-67).

9. With regard to claim 15, Becker-Szendy and Kazar disclose,

- *wherein the multi-protocol server is further adapted to process data access requests in response to one or more file-level protocols.* (Becker-Szendy, col.1, line 10 – col.20, line 48; Kazar, col.1, line 7 – col.11, line 19)

Becker-Szendy discloses, *"The present invention satisfies this need, and presents a system, a computer program product, and an associated method (collectively referred to herein as "the system" or "the present system") and a service for federating a local file system into a distributed file system (FS), while preserving local access to the existing data in the local file system. The present system may provide indirect access to local file systems using protocols such as, for example, storage tank protocols, object-based storage protocols, block-based protocols, etc. The server-based design of the present system allows systems to migrate their data and share the management of data. The data is federated, or made available to various clients by making it on-line to each client. The present system may be used with any file system protocol that supports migration, consistency and multi-host federation"* (Becker-Szendy, col.2, lines 49-64).

10. Claims 4-5 and 18-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Becker-Szendy et al. (US007243089B2), in view of Kazar et al. (US006868417B2), and further in view of Mane et al. (US20050050107A1).

11. With regard to claims 4-5 and 18-19, Becker-Szendy and Kazar disclose,

See *claims 3 and 17* rejection as detailed above.

However, Becker-Szendy and Kazar do not explicitly disclose,

- *wherein the units of storage resources comprise volumes.*
- *wherein the units of storage resources comprise qtrees.*

Mane teaches,

- *wherein the units of storage resources comprise volumes.* (Mane, para.1-53)

Mane discloses, "*The processor 25 includes a number of program layers, including a network interface 26 for coupling to the data network, a file system layer 27 for organizing data into a hierarchical file system of files and directories, a volume layer 28 for organizing the data into logical volumes of data blocks, and a Small Computer System Interface (SCSI) driver 29 for linking the volume layer 28 to the disk storage 24*" (Mane, para.25). Hence, Mane teaches of logical volumes of data blocks (i.e., Applicant's volumes) as units of storage resources.

- *wherein the units of storage resources comprise qtrees* (Mane, para.1-53)

Mane discloses, "*In accordance with another aspect, the invention provides a method of maintaining quotas for storage resources used by a file server for storing files in selected directory trees of a file system. The file server has a tree quota database of usage values of the storage resources and limit values for the storage resources for the selected directory trees of the file system*" (Mane,

para.6). Hence, Mane teaches of a tree quota database (i.e., Applicant's qtree) as a unit of storage resources.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of Mane with the teachings of Becker-Szendy and Kazar to "[provide] a method of maintaining quotas for storage resources used by a file server for storing files in selected directory trees of a file system" (Mane, para.5). Mane discloses, "A preferred way of preventing the renaming of a file from causing an undesired nesting of quota trees is to prevent the renaming of a file from causing a file to be moved into, out of, or between quota trees. In the preferred implementation, the quota are treated as if they were separate file systems by returning a cross-device error if the renaming of a file would otherwise cause a file to be moved into, out of, or between quota trees" (Mane, para.48). Becker-Szendy discloses, "What is therefore needed is a system, a service, a computer program product, and an associated method for federating an old system into a new system, and optionally migrating data from an old system to a new system. This method should operate seamlessly and efficiently with minimum disruption to existing applications running on the system. Further, this method should ensure data consistency for existing applications while making the data available for migration in a federated system. The need for such a solution has heretofore remained unsatisfied" (Becker-Szendy, col.2, lines 35-44).

12. Claims 7-11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Becker-Szendy et al. (US007243089B2), in view of Kazar et al. (US006868417B2), and further in view of George et al. (US007010663B2).

13. With regard to claim 7, Becker-Szendy and KAzar disclose,

See claim 6 rejection as detailed above.

However, Becker-Szendy and Kazar do not explicitly disclose,

- *wherein the operating system further comprises a user interface having a command set adapted to operate on virtual disks, and wherein the command set executes within a context of a vfiler.*

George teaches,

- *wherein the operating system further comprises a user interface having a command set adapted to operate on virtual disks, and wherein the command set executes within a context of a vfiler. (George, col.1, line 8 – col.14, line 58)*

George discloses, "Referring now to FIG. 2, the data storage device 120 of FIG. 1 is shown that provides for managing a plurality of virtual LUNs over one or more existing volumes of storage within the storage device 120, in accordance with one embodiment of the present invention. The data storage device 120 comprises two interfaces for receiving and sending command line interface (CLI) instructions and Input/Output (I/O) data. The interfaces include a CLI interface and a hypertext transfer protocol (HTTP) interface" (George, col.5, lines 33-41). Hence, George teaches of two interfaces (i.e., Applicant's user interface) for receiving and sending instructions (i.e., Applicant's having a command set) for managing (i.e., Applicant's adapted to operate on) a plurality of virtual LUNs (i.e., Applicant's virtual disks) over one or more existing volumes of storage via the virtualization layer (i.e., Applicant's within a context of a vfiler).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of George with the teachings of Becker-Szendy Kazar to provide a method for *"partitioning the existing volumes into a plurality of slices. Each of the plurality of slices is then mapped to the plurality of virtual LUNs. Furthermore, each of the plurality of virtual LUNs is masked to each of the plurality of host applications to provide access control. Moreover, the plurality of host applications are transparently interfaced with the existing volumes via a virtualization software layer that interfaces with and preserves the originally configured internal intelligence (e.g., internal operating code) that accesses the plurality of volumes"* (George, col.2, lines 39-49). Georges discloses, *"Embodiments of the present invention relate to the field of data storage systems. More particularly, embodiments of the present invention relate generally to the expansion of an existing data storage system into a plurality of virtual data storage systems"* (George, col.1, lines 9-13). Becker-Szendy discloses, *"What is therefore needed is a system, a service, a computer program product, and an associated method for federating an old system into a new system, and optionally migrating data from an old system to a new system. This method should operate seamlessly and efficiently with minimum disruption to existing applications running on the system. Further, this method should ensure data consistency for existing applications while making the data available for migration in a federated system. The need for such a solution has heretofore remained unsatisfied"* (Becker-Szendy, col.2, lines 35-44).

14. With regard to claims 8-9, Becker-Szendy, Kazar, and George disclose,

- *wherein the user interfaces comprises a command line interface (CLI) adapted to support the command set. (Becker-Szendy, col.1, line 10 – col.20, line 48; Kazar, col.1, line 7 – col.11, line 19; George, col.1, line 8 – col.14, line 58)*

George discloses, "Referring now to FIG. 2, the data storage device 120 of FIG. 1 is shown that provides for managing a plurality of virtual LUNs over one or more existing volumes of storage within the storage device 120, in accordance with one embodiment of the present invention. The data storage device 120 comprises two interfaces for receiving and sending command line interface (CLI) instructions and Input/Output (I/O) data. The interfaces include a CLI interface and a hypertext transfer protocol (HTTP) interface" (George, col.5, lines 33-41).

- *wherein the CLI comprises a lun command adapted to perform operations to a virtual disk associated with the context of the vfiler. (Becker-Szendy, col.1, line 10 – col.20, line 48; Kazar, col.1, line 7 – col.11, line 19; George, col.1, line 8 – col.14, line 58)*

George discloses, "Typically, the CLI interface provides access by a user (e.g., system administrator) to configure, update, and/or modify the data storage device 120, such as, creating or removing virtual LUNs, and expanding or reducing the size of virtual LUNs, etc. In FIG. 2, the CLI interface is provided through port task 215 that functions essentially for properly routing the CLI instructions through storage device 120. In another embodiment, the HTTP interface, through port task 210, also allows access by a user to configure the storage device 120. In addition, the HTTP interface, through port task 210, provides for an avenue for access by other users and host applications to the data storage device 120, as will be discussed" (George, col.5, lines 42-54).

15. With regard to claims 10-11, Becker-Szendy, Kazar, and George disclose,

- *wherein the lun command creates a logical unit number on a file system associated with the server, the logical unit number being associated with the context of the vfiler.* (Becker-Szendy, col.1, line 10 – col.20, line 48; Kazar, col.1, line 7 – col.11, line 19; George, col.1, line 8 – col.14, line 58)

George discloses, "This disclosure describes a method and apparatus for slicing one or more volumes of storage into a plurality of virtual LUNs, thereby increasing the number of accessible LUNs within a data storage network. Also, another embodiment of the present invention discloses a method and system for increasing the number of host applications that can access and use a particular volume of storage" (George, col.4, lines 11-17).

- *wherein the CLI comprises an igroup command that generates a set of file system primitive for binding an initiator group to one or more initiator addresses and wherein the initiator group is associated with the context of the virtual server.* (Becker-Szendy, col.1, line 10 – col.20, line 48; Kazar, col.1, line 7 – col.11, line 19; George, col.1, line 8 – col.14, line 58)

Response to Arguments

16. Applicant's arguments with respect to *claims 1, 16, 20 and 24-26* have been considered but they are not persuasive.

17. With regard to claims 1, 16, 20 and 24-26, the Applicant points out that:

- *Accordingly, Applicant respectfully urges that the cited Becker-Szendy and Kazar patents, either taken singly or in combination, are legally incapable of rendering Applicant's claimed invention unpatentable under 35 U.S.C. 103(a) because of the absence from each of Applicant's claimed a plurality of virtual servers . . . a context data structure provided to each virtual server, the context data structure including information pertaining to a security domain of the virtual server for each supported access protocol, to enable controlled access to the shared resources of the file system.*

However, the Examiner finds that the Applicants' arguments are not persuasive because Becker-Szendy discloses, *"File data is the bytes that are actually stored in a file. Metadata is all the rest of the information stored in a file system. Metadata comprises the directory tree and the attributes of objects such as files and directories. The directory tree is a set of names that are arranged in directories, forming a tree structure. Typical attributes comprise time stamps (i.e., time created, time last modified, time last read) and security related attributes (i.e., the identity of the owner of the object and a description of what the owner or other parties may do to the object). In addition, with object based storage, it is possible to have some of the metadata stored on the object based storage. For example, object based storage manages the block mapping internally, off-loading this role from the file system"* (Becker-Szendy, col.7, lines 41-54).

Conclusion

18. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a). A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.
19. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Thomas Duong whose telephone number is 571/272-3911. The examiner can normally be reached on M-F 7:30AM - 4:00PM. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Vivek Srivastava can be reached on 571/272-7304. The fax phone numbers for the organization where this application or proceeding is assigned are 571/273-8300 for regular communications and 571/273-8300 for After Final communications.

/Thomas Duong/

Patent Examiner, Art Unit 2445

Art Unit: 2445

July 7, 2009

/NIVEK SRIVASTAVA/

Supervisory Patent Examiner, Art Unit 2445